So what can we do with statistics in society?

Nnenna Asidianya

University of Toronto

Tidyverse is collection of R packages designed for data analysis.

Figure: https://www.tidyverse.org/

Installation code:

```
install.packages("tidyverse")
```

ggplot2: mtcars data

In order to understand why tidyverse is important we need to have a working framework of what similar manipulations look like in R.

mtcars data

- From the 1974 Motor Trend US magazine.
- Presents fuel consumption and 10 aspects of automobile design and performance for 32 automobiles from 1973–74 models

ggplot2: mtcars data

Here is the data...

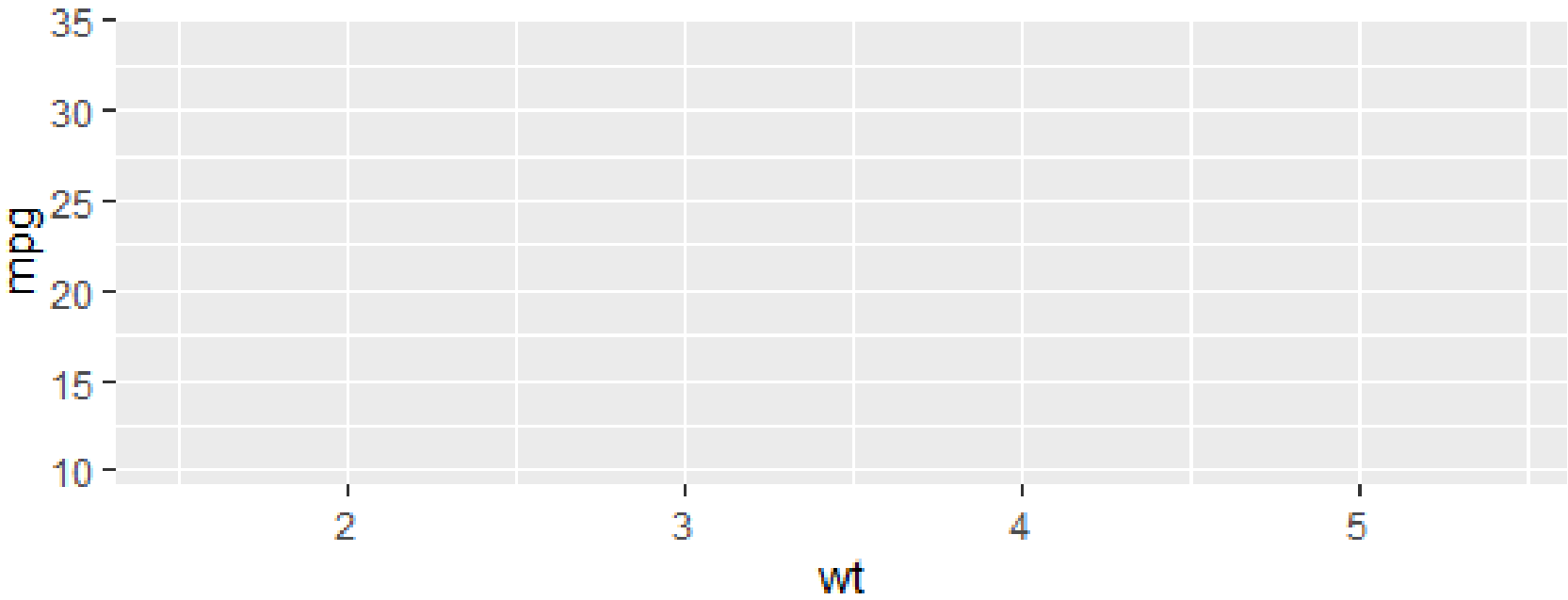
```
head(CARS)
                   mpg  cyl disp  hp  drat
Mazda RX4          21.0   6  160 110 3.90 2.620
Mazda RX4 Wag      21.0   6  160 110 3.90 2.875
Datsun 710         22.8   4  108  93 3.85 2.320
Hornet 4 Drive     21.4   6  258 110 3.08 3.215
Hornet Sportabout 18.7   8  360 175 3.15 3.440
Valiant            18.1   6  225 105 2.76 3.460
                   qsec vs am gear carb
Mazda RX4          16.46  0  1    4    4
Mazda RX4 Wag      17.02  0  1    4    4
Datsun 710         18.61  1  1    4    1
Hornet 4 Drive     19.44  1  0    3    1
Hornet Sportabout 17.02  0  0    3    2
Valiant            20.22  1  0    3    1
>
```

ggplot2: mtcars data

ggplot2 is a data visualization package contained inside Tidyverse

Question: Understand the relationship between weight (wt) and miles per gallon (mpg)

```
ggplot(CARS, aes(x=wt, y=mpg))
```

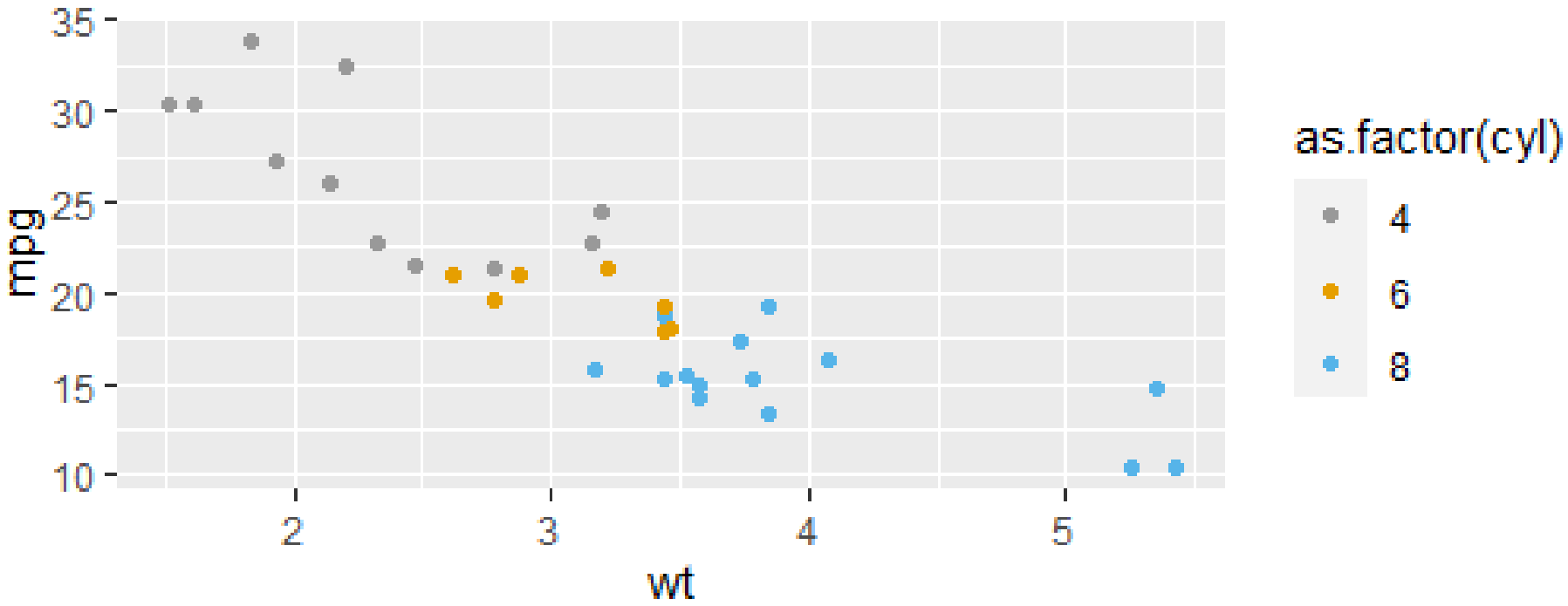


ggplot2: mtcars data

Question: Understand the relationship between weight (wt) and miles per gallon (mpg)

- Additional

`Demo in R.`



readr: Ontario COVID cases

readr provides an easy way to read rectangular data in R (i.e. each column is a variable and each row is a case or record.)

- `read_csv`: reads comma separated (CSV) files
- `read_table` reads tabular files where columns are separated by white-space (i.e. txt files).

Recall: In Base R we have the following functions:

- `read.csv`
- `read.table`

readr: Ontario COVID cases

The `read_csv` function imports data into R as a tibble, while `read.csv` imports an R data frame. Here are **some** interesting features:

- Tibbles load faster.
- Tibbles don't create row names (example to follow)
- Allow for non-standard variable names (i.e. numerical values or spaces in your variables)

readr: Ontario COVID cases

"ages.csv:" Aggregated data of the active cases, resolved cases, and deaths from COVID up to February 21, 2021:

```
# A tibble: 6 x 5
category Active Resolved Deaths X5
<chr>      <dbl>    <dbl>  <dbl> <chr>
1 Under 20   1203     6958      1 20-Nov
2 Under 20   1584    35577      2 21-Feb
3 20-29      1662    14419      4 20-Nov
4 20-29      2406    56567      8 21-Feb
5 30-39      1248    10784      7 20-Nov
6 30-39      2128    42528     18 21-Feb
>
```

- Maintains the input types (i.e. <dbl> = double precision class; recognizes decimal places)
- **Question:** What happens if the first row is not variable labels? (demo)

tidyr: Ontario COVID cases

tidyr: As it's namesake implies, it is an efficient way of ensuring that the data is tidy. These are common in repeated measures studies.

- `pivot_wider()`
- `pivot_longer()`
- `gather()`; can convert to wide or long format (easier when your "key" has more than two levels)

tidyr: Ontario COVID cases

Question: How to create a data splits active cases, recovered cases, and deaths per age group?

- Note:the current data is in long format.

We need to use the `pivot_wider()` function in tidyr (demo)!

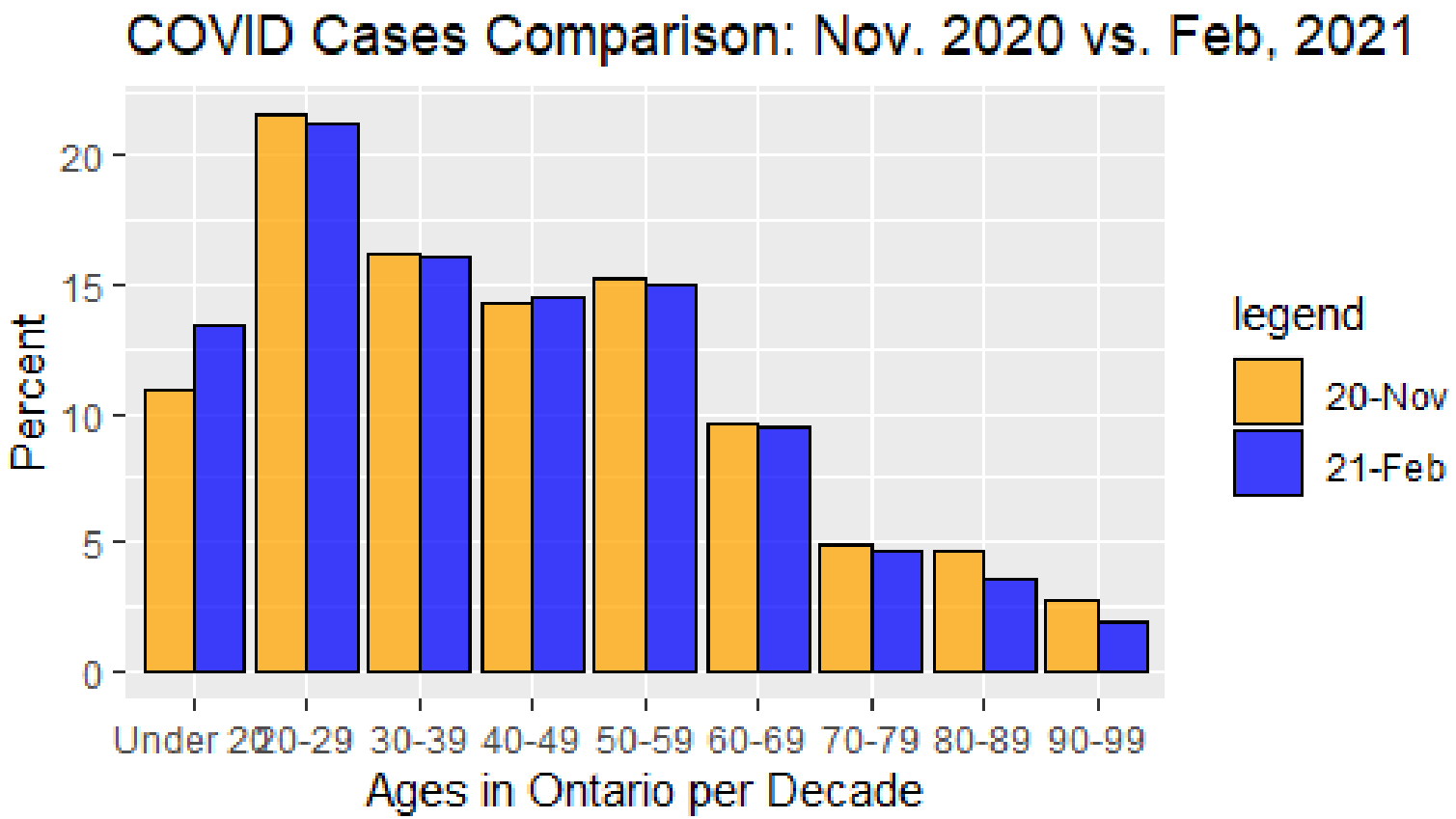
dplyr: Ontario COVID cases

dplyr: is a package for data manipulation. Can solve the most common data manipulation challenges in an efficient manner:

- `mutate()` adds new variables that are functions of existing variables
- `select()` picks variables based on their names.
- `filter()` picks cases based on their values.
- `summarise()` reduces multiple values down to a single summary.
- `group_by()` performs an operation across groups.

dplyr: Ontario COVID cases

Question 1: How does the Ontario COVID cases compare across ages at the different timepoints?



Question 2: How did I get here? How was this data processed? (Demo)

dplyr: Ontario COVID cases

```
df<-ages %>% mutate(category = fct_relevel(category,
"Under 20", "20-29", "30-39",
"40-49", "50-59", "60-69", "70-79","80-89",
"90-99", "Unknown")) %>%  filter(!category=="Unknown")
%>% group_by(Date) %>%
mutate(total_cases=Active+Resolved) %>%
mutate(pct_cases=100*(total_cases/sum(total_cases)))
```

- `mutate()` was used to re-order age categories in ascending years rather than alphabetical order.
- `filter()` removed rows that did not have known ages.
- `group_by` was used to aggregate total cases based on date.

Thanks for Watching!